The Flaw at the Heart of the Internet

DAN KAMINSKY DISCOVERED A FUNDAMENTAL SECURITY PROBLEM IN THE INTERNET AND GOT PEOPLE TO CARE IN TIME TO FIX IT. IT'S A DRAMATIC STORY WITH A HAPPY ENDING ... BUT WE WERE LUCKY THIS TIME.

By ERICA NAONE

Dan Kaminsky, uncharacteristically, was not looking for bugs earlier this year when he happened upon a flaw at the core of the Internet. The security researcher was using his knowledge of Internet infrastructure to come up with a better way to stream videos to users. Kaminsky's expertise is in the Internet's domain name system (DNS), the protocol responsible for matching websites' URLs with the numeric addresses of the servers that host them. The same content can be hosted by multiple servers with several addresses, and Kaminsky thought he had a great trick for directing users to the servers best able to handle their requests at any given moment.

Normally, DNS is reliable but not nimble. When a computer—say, a server that helps direct traffic across Comcast's network—requests the numerical address associated with a given URL, it stores the answer for a period of time known as "time to live," which can be anywhere from seconds to days. This helps to reduce the number of requests the server makes. Kaminsky's idea was to bypass the time to live, allowing the server to get a fresh answer every time it wanted to know a site's address. Consequently, traffic on Comcast's network would be sent to the optimal address at every moment, rather than to whatever address had already been stored. Kaminsky was sure that the strategy could significantly speed up content distribution.

It was only later, after talking casually about the idea with a friend, that Kaminsky realized his "trick" could completely break the security of the domain name system and, therefore, of the Internet itself. The time to live, it turns out, was at the core of DNS security; being able to bypass it allowed for a wide variety of attacks. Kaminsky wrote a little code to make sure the situation was as bad as he thought it was. "Once I saw it work, my stomach dropped," he says. "I thought, 'What the heck am I going to do about this? This affects everything.'"

Kaminsky's technique could be used to direct Web surfers to any Web page an attacker chose. The most obvious use is to send people to phishing sites (websites designed to trick people into entering banking passwords and other personal information, allowing an attacker to steal their identities) or other fake versions of Web pages. But the danger is even worse: protocols such as those used to deliver e-mail or for secure communications over the Internet ultimately rely on DNS. A creative attacker could use Kaminsky's technique to intercept sensitive e-mail, or to create forged versions of the certificates that ensure secure transactions between users and banking websites. "Every day I find another domino," Kaminsky says. "Another thing falls over if DNS is bad. ... I mean, literally, you look around and see anything that's using a network—*anything* that's using a network—and it's probably using DNS."

Kaminsky called Paul Vixie, president of the Internet Systems Consortium, a nonprofit corporation that supports several aspects of Internet infrastructure, including the software most commonly used in the domain name system. "Usually, if somebody wants to report a problem, you expect that it's going to take a fair amount of time for them to explain it—maybe a whiteboard, maybe a Word document or two," Vixie says. "In this case, it took 20 seconds for him to explain the problem, and another 20 seconds for him to answer my objections. After that, I said, 'Dan, I am speaking to you over an unsecure cell phone. Please do not ever say to anyone what you just said to me over an unsecure cell phone again.'"

Perhaps most frightening was that because the vulnerability was not located in any particular hardware or software but in the design of the DNS protocol itself, it wasn't clear how to fix it. In secret, Kaminsky and Vixie gathered together some of the top DNS experts in the world: people from the U.S. government and

Photograph by JOHN KEATLEY

high-level engineers from the major manufacturers of DNS software and hardware—companies that include Cisco and Microsoft. They arranged a meeting in March at Microsoft's campus in Redmond, WA. The arrangements were so secretive and rushed, Kaminsky says, that "there were people on jets to Microsoft who didn't even know what the bug was."

Once in Redmond, the group tried to determine the extent of the flaw and sort out a possible fix. They settled on a stopgap measure that fixed most problems, would be relatively easy to deploy, and would mask the exact nature of the flaw. Because attackers commonly identify security holes by reverse-engineering patches intended to fix them, the group decided that all its members had to release the patch simultaneously (the release date would turn out to be July 8). Kaminsky also asked security researchers not to publicly speculate on the details of the flaw for 30 days after the release of the patch, in an attempt to give companies enough time to secure their servers.

On August 6, at the Black Hat conference, the annual gathering of the world's Internet security experts, Kaminsky would publicly reveal what the flaw was and how it could be exploited.

ASKING FOR TROUBLE

Kaminsky has not really discovered a new attack. Instead, he has found an ingenious way to breathe life into a very old one. Indeed, the basic flaw targeted by his attack predates the Internet itself.

The foundation of DNS was laid in 1983 by Paul Mockapetris, then at the University of Southern California, in the days of ARPAnet, the U.S. Defense Department research project that linked computers at a small number of universities and research institutions and ultimately led to the Internet. The system is designed to work like a telephone company's 411 service: given a name, it looks up the numbers that will lead to the bearer of that name. DNS became necessary as ARPAnet grew beyond an individual's ability to keep track of the numerical addresses in the network.

Mockapetris, who is now chairman and chief scientist of Nominum, a provider of infrastructure software based in Redwood, CA, designed DNS as a hierarchy. When someone types the URL for a Web page into a browser or clicks on a hyperlink, a request goes to a name server maintained by the user's Internet service provider (ISP). The ISP's server stores the numerical addresses of URLs it handles frequently—at least, until their time to live expires. But if it can't find an address, it queries one of the 13 DNS root servers, which directs the request to a name server responsible for one of the top-level domains, such as .com or .edu. That server forwards the request to a server specific to a single domain name, such as google.com or mit.edu. The forwarding continues through servers with ever more specific responsibilities—mail.google.com, or libraries.mit.edu—until the request reaches a server that can either give the numerical address requested or respond that no such address exists.

As the Internet matured, it became clear that DNS was not secure enough. The process of passing a request from one server to the next gives attackers many opportunities to intervene with false responses, and the system had no safeguards to ensure that the name server answering a request was trustworthy. As early as 1989, Mockapetris says, there were instances of "cache poisoning," in which a name server was tricked into storing false information about the numerical address associated with a website.

In the 1990s, the poisoner's job was relatively easy. The lower-level name servers are generally maintained by private entities: Amazon, for instance, controls the addresses supplied by the amazon.com name server. If a low-level name server can't find a requested address, it will either refer the requester to another name server or tell the requester the page doesn't exist. But in the '90s, the low-level server could also furnish the requester with the top-level server's address. To poison a cache, an attacker simply had to falsify that information. If an attacker tricked, say, an ISP's name server into storing the wrong address for the .com server, it could hijack most of the traffic traveling over the ISP's network.

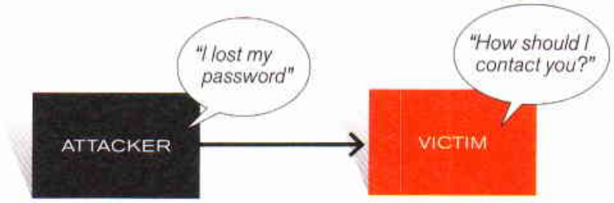
Mockapetris says several features were subsequently added to DNS to protect the system. Requesting servers stopped accepting higher-level numerical addresses from lower-level name servers. But attackers found a way around that restriction. As before, they would refer a requester back to, say, the .com server. But now the requester had to look up the .com server's address on its own. It would request the address, and the attacker would race to respond with a forged reply before the real reply arrived. Ad hoc security measures were added to protect against this strategy, too. Now, each request to a DNS server carries a randomly generated transaction ID, one of 65,000 possible numbers, which the reply must contain as well. An attacker racing to beat a legitimate reply would also have to guess the correct transaction ID. Unfortunately, a computer can generate so many false replies so quickly that if it has enough chances, it's bound to find the correct ID. So the time to live, originally meant to keep name servers from being overburdened by too many requests, became yet another stopgap security feature. Because the requesting server will store an answer for some period of time, the attacker gets only a few chances to attempt a forgery. Most of the time, when the server needs a .com address, it consults its cache rather than checking with the .com server.

Kaminsky found a way to bypass these ad hoc security features—most important, the time to live. That made the system just as vulnerable as it was when cache poisoning was first discovered. Using Kaminsky's technique, an attacker gets a nearly infinite number of chances to supply a forgery.

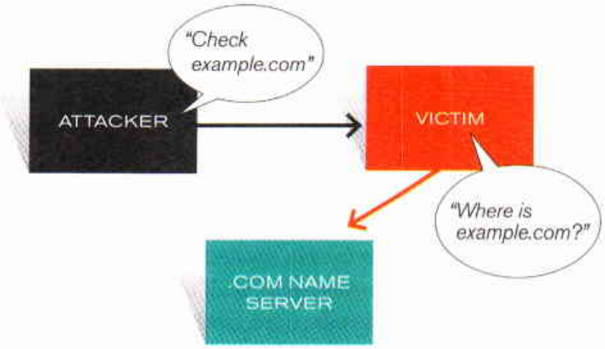
Say an attacker wants to hijack all the e-mail that a social-networking site like Facebook or MySpace sends to Gmail accounts. He signs up for an account with the social network, and

A CACHE-POISONING ATTACK

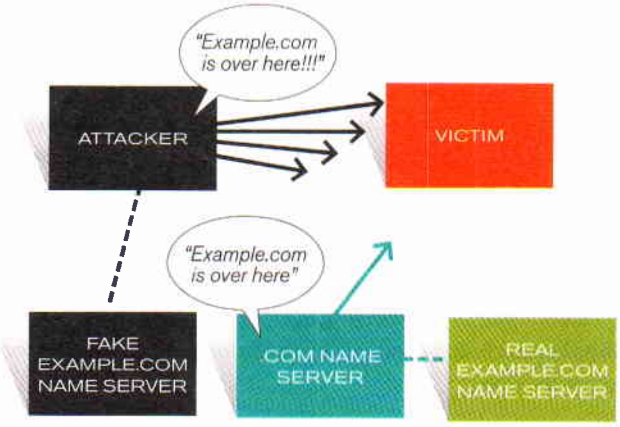
Cache poisoning causes a requesting server to store false information about the numerical address associated with a website. A basic version of the attack–without some of the more sophisticated techniques Kaminsky employs–is outlined below.



1. To begin, the attacker lures the victim's server into contacting a domain the attacker controls. The attacker could, say, claim to have forgotten a password, prompting the victim to respond by e-mail.



2. The victim performs a DNS lookup to find out where to send the e-mail. But the attacker's name server refers the victim to another server, such as that of example.com. Since the attacker knows that the victim will now start a DNS lookup for that server, he or she has an opportunity to attempt to poison its cache.


"Example.com is over here!!!"
ATTACKER
VICTIM
"Example.com is over here"
FAKE EXAMPLE.COM NAME SERVER
.COM NAME SERVER
REAL EXAMPLE.COM NAME SERVER

3. The attacker tries to supply a false response before the legitimate server can supply the real one. If the attacker guesses the right ID number, the victim accepts the false reply, which poisons the cache.

when he's prompted for an e-mail address, he supplies one that points to a domain he controls. He begins to log on to the social network but claims to have forgotten his password. When the system tries to send a new password, it does a DNS lookup that leads to the attacker's domain. But the attacker's server claims that the requested address is invalid.

At this point, the attacker could refer the requester to the google.com name servers and race to supply a forged response. But then he would get only one shot at cracking the transaction ID. So instead, he refers the requester to the nonexistent domains 1.google.com, then 2.google.com, then 3.google.com, and so on, sending a flood of phony responses for each. Each time, the requesting server will consult Google's name servers rather than its cache, since it won't have stored addresses for any of the phony URLs. The attack completely bypasses the limits set by the time to live. One of the attacker's forgeries is bound to get through. Then it's a simple matter to direct anything the requesting server intends for Google to the attacker's own servers, since the attacker appears to have authority for URLs ending in google.com. Kaminsky says he was able to pull off test attacks in as little as 10 seconds.

IN THE DARK

On July 8, Kaminsky held the promised press conference, announcing the release of the patch and asking other researchers not to speculate on the flaw. The hardware and software vendors had settled on a patch that forces an attacker to guess a longer transaction ID. Kaminsky says that before the patch, the attacker had to make tens of thousands of attempts to successfully poison a cache. After the patch, it would have to make billions.

News of the flaw appeared in the *New York Times*, on the BBC's website, and in nearly every technical publication. Systems administrators scrambled to get the patch worked into their systems before they could be attacked. But because Kaminsky failed to provide details of the flaw, some members of the security community were skeptical. Thomas Ptacek, a researcher at Matasano Security, posted on Twitter: "Saying it here first: doubting there's really any meat to this DNS security announcement."

Dino Dai Zovi, a security researcher best known for finding ways to deliver malware to a fully patched Macbook Pro, says, "I was definitely skeptical of the nature of the vulnerability, especially because of the amount of hype and attention versus the low amount of details. Whenever I see something like that, I instantly put on my skeptic hat, because it looks a lot like someone with a vested interest rather than someone trying to get something fixed." Dai Zovi and others noted that the timing was perfect to promote Kaminsky's Black Hat appearance, and they bristled at the request to refrain from speculation.

The lack of information was particularly controversial because system administrators are often responsible for evaluating patches and deciding whether to apply them, weighing the danger of the security flaw against the disruption that the patch will cause. Because DNS is central to the operation of any Internet-dependent organization, altering it isn't something that's done lightly. To make matters worse, this patch didn't work properly with certain types of corporate firewalls. Many IT professionals expressed frustration at the lack of detail, saying that they were unable to properly evaluate the patch when so much remained hidden.

Concerned by the skepticism about his claims, Kaminsky held a conference call with Ptacek and Dai Zovi, hoping to make them see how dangerous the bug was. Both came out of the call converted. But although Dai Zovi notes that much has changed since the time when hardware and software manufacturers dealt with flaws by simply denying that security researchers had identified real problems, he also says, "We don't know what to do when the vulnerabilities are in really big systems like DNS." Researchers face a dilemma, he says: they need to explain flaws in order to convince others of their severity, but a vulnerability like the one Kaminsky found is so serious that revealing its details might endanger the public.

Halvar Flake, a German security researcher, was one observer who thought that keeping quiet was the more harmful alternative. Public speculation is just what's needed, he says, to help people understand what could hit them. Flake read a few basic materials, including the German Wikipedia entry on DNS, and wrote a blog entry about what he thought Kaminsky might have found. Declaring that his guess was probably wrong, he invited other researchers to correct him. Somehow, amid the commotion his post caused in the security community, a detailed explanation of the flaw appeared on a site hosted by Ptacek's employer, Matasano Security. The explanation was quickly taken down, but not before it had proliferated across the Internet.

Chaos ensued. Kaminsky posted on Twitter, "DNS bug is public. You need to patch, or switch to [Web-based] OpenDNS, RIGHT NOW." Within days, Metasploit, a computer security project that designs sample attacks to aid in testing, released two modules exploiting Kaminsky's flaw. Shortly after, one of the first attacks based on the DNS flaw was seen in the wild. It took over some of AT&T's servers in order to present a false Google home page, loaded with the attacker's own ads.

OUT OF COOKIES

Thirty minutes before Kaminsky took the stage at Black Hat to reveal the details of the flaw at last, people started to flood the ballroom at Caesar's Palace in Las Vegas. The speaker preceding Kaminsky hastened to wrap things up. Seats ran out, and people sat cross-legged on every square inch of carpet. Kaminsky's grandmother, who was sitting in the front row, had baked 250 cookies for the event. There were nowhere near enough.

Depending on your perspective, the way Kaminsky handled the DNS flaw and its patch was either dangerous grandstanding that left many Internet users vulnerable or a "media hack" necessary to train a spotlight on the bug's dangers.

Kaminsky walked up to the podium. "There's a lot of people out there," he said. "Holy crap." Kaminsky is tall, and his gestures are a little awkward. As of early August, he said, more than 120 million broadband customers had been protected, as Internet service providers applied patches. Seventy percent of Fortune 500 companies had patched their systems, and an additional 15 percent were working on it. However, he added, 30 to 40 percent of name servers on the Internet were still unpatched and vulnerable to his 10-second cache-poisoning attack.

Onstage, he flipped between gleeful description of his discovery's dark possibilities and attempts to muster the seriousness appropriate to their gravity. He spoke for 75 minutes, growing visibly lighter as he unburdened himself of seven months' worth of secrets. As he ended his talk, the crowd swept close to him, and he was whisked off by reporter after reporter.

Even those security experts who agreed that the vulnerability was serious were taken aback by Kaminsky's eager embrace of the media attention and his relentless effort to publicize the flaw. Later that day, Kaminsky received the Pwnie award for "most overhyped bug" from a group of security researchers. (The word "pwn," which rhymes with "own," is Internet slang for "dominate completely." Kaminsky's award is subtitled "The Pwnie for pwning the media.") Dai Zovi, presenting the award, tried to list the publications that had carried Kaminsky's story. He gave up, saying, "What *weren't* you in?"

"*GQ*!" someone shouted from the audience.

Kaminsky took the stage and spat out two sentences: "Some people find bugs; some people get bugs fixed. I'm happy to be in the second category." Swinging the award—a golden toy pony—by its bright pink hair, he stalked down the long aisle of the ballroom and out the door.

WHO'S IN CHARGE?

Depending on your perspective, the way Kaminsky handled the DNS flaw and its patch was either dangerous grandstanding that needlessly called public attention to the Internet vulnerability or—as Kaminsky sees it—a "media hack" necessary to train a spotlight on the bug's dangers. Either way, the story points to the troubling absence of any process for identifying and fixing critical flaws in the Internet. Because the Internet is so decentralized, there simply isn't a specific person or organization in charge of solving its problems.

And though Kaminsky's flaw is especially serious, experts say it's probably not the only one in the Internet's infrastructure. Many Internet protocols weren't designed for the uses they're put to today; many of its security features were tacked on and don't address underlying vulnerabilities. "Long-term, architecturally, we have to stop assuming the network is as friendly as it is," Kaminsky says. "We're just addicted to moving sensitive information across the Internet insecurely. We can do better."

Indeed, at another security conference just days after Kaminsky's presentation at Black Hat, a team of researchers gave a talk illustrating serious flaws in the Internet's routing border gateway protocol. Like Kaminsky, the researchers had found problems with the fundamental design of an Internet protocol. Like the DNS flaw, the problem could allow an attacker to get broad access to sensitive traffic sent over the Internet.

Many experts say that what happened with the DNS flaw represents the best-case scenario. Mischel Kwon, director of US-CERT, a division of the Department of Homeland Security that helped get out the word about the DNS bug, hopes the network of organizations that worked together in this case will do the same if other flaws emerge. Though there's no hierarchy of authority in the private sector, Kwon says, there are strong connections between companies and organizations with the power to deploy patches. She says she is confident that, considering the money and effort being poured into improving security on the Internet, outdated protocols will be brought up to date.

But that confidence isn't grounded in a well-considered strategy. What if Kaminsky hadn't had extensive connections within the security community or, worse, hadn't been committed to fixing the flaw in the first place? What if he had been a true "black hat" bent on exploiting the vulnerability he'd discovered? What if his seemingly skillful manipulation of the media had backfired, and the details of the flaw had become known before the patch was in place?

What's more, even given the good intentions of researchers like Kaminsky, fixing basic flaws in the Internet isn't easy. Experts agree that the DNS problem is no exception. Several proposals are on the table for solving it by means more reliable than a patch, mostly by reducing the trust a requesting server accords a name server. Proposals range from relatively simple fixes, such as including even more random information in the requests made to name servers, to moving the entire system over to a set of protocols that would let name servers sign their responses cryptographically.

In the meantime, both Kaminsky and Vixie say attackers have started to make use of the DNS flaw, and they expect more trouble to come. Kaminsky notes that the flaw becomes particularly dangerous when exploited along with other vulnerabilities. One such combination, he says, would allow an attacker to take over the automatic updates that a software vendor sends its customers, replacing them with malware. Kaminsky says he's spent the last several months on the phone to companies that would be attractive targets for that kind of attack, such as certificate authorities, social networks, and Internet service providers, trying to convince them to patch as soon as possible.

"The scary thing," Dai Zovi says, "is how fragile [the Internet] is. ... And what are we going to do about it?"

ERICA NAONE IS AN ASSISTANT EDITOR AT *TECHNOLOGY REVIEW*.